

IE 410 – INTRODUCTION TO ROBOTICS

Lab-2 report

Team M410:

201901011 – HIMANSHU DUDHATRA

201901024 – DHAVALSINH RAJ

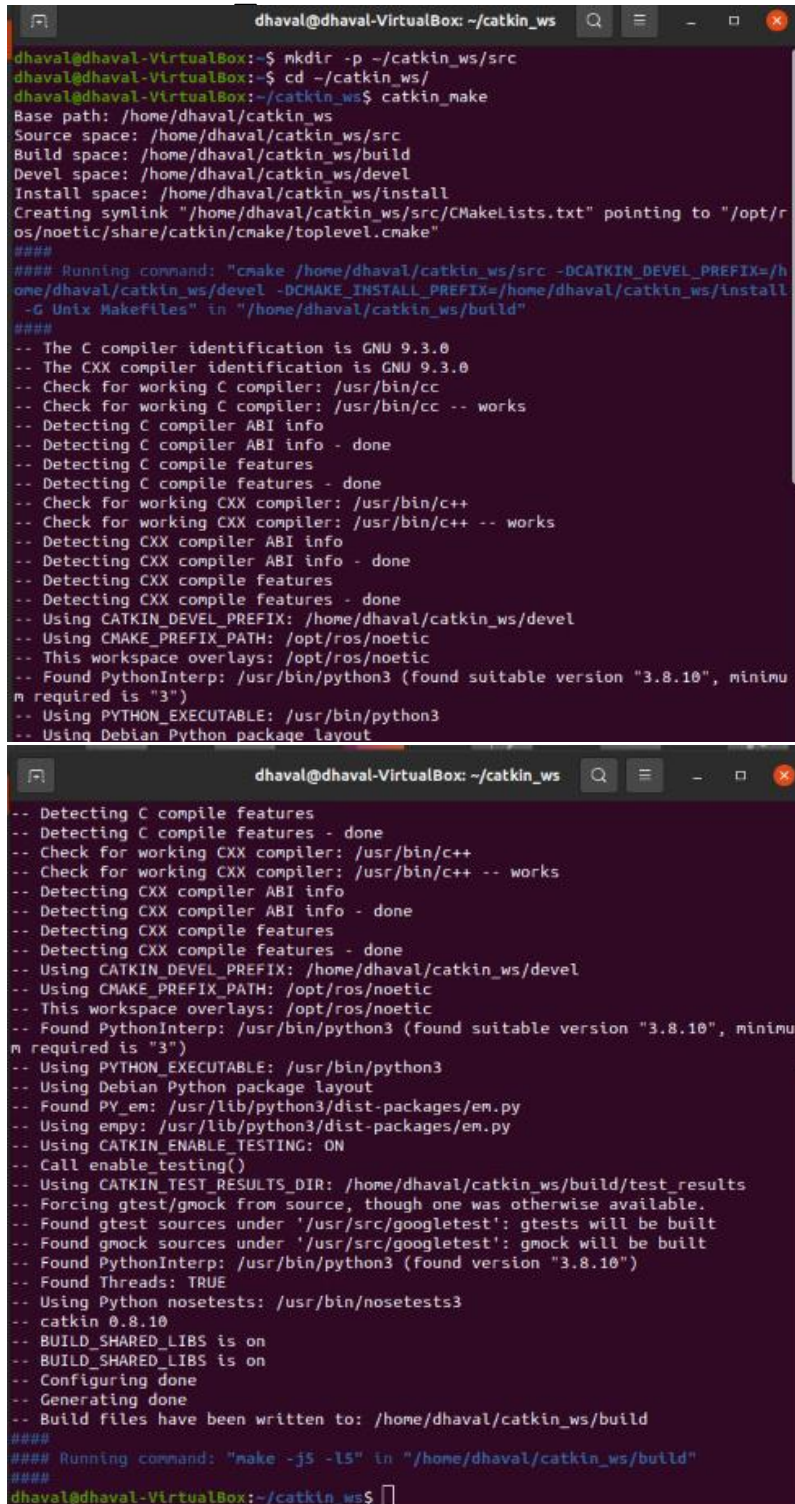
201901100 – SHUBHAM PATEL

201901145 – GARGEY PATEL

- In this lab, we will use ROS and python to use Turtlesim.
- We will use following commands for turtlesim.
- Creating ROS workspace

Commands to create and build a catkin workspace:

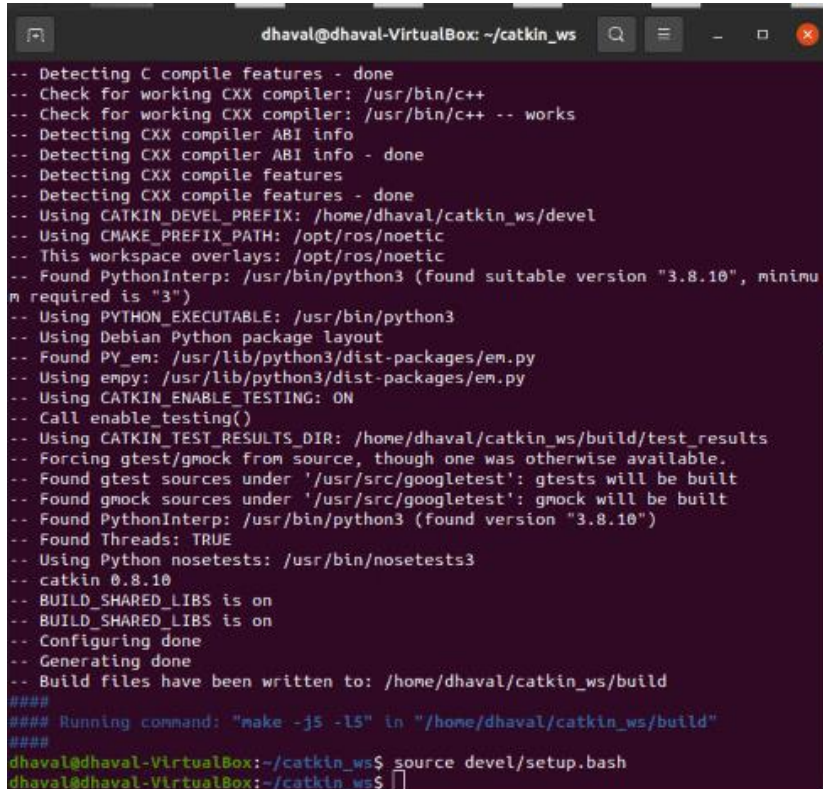
```
$ mkdir -p ~/catkin_ws/src
$ cd ~/catkin_ws/
$ catkin make
```



```
dhaval@dhaval-VirtualBox: ~/catkin_ws
dhaval@dhaval-VirtualBox:~$ mkdir -p ~/catkin_ws/src
dhaval@dhaval-VirtualBox:~$ cd ~/catkin_ws/
dhaval@dhaval-VirtualBox:~/catkin_ws$ catkin_make
Base path: /home/dhaval/catkin_ws
Source space: /home/dhaval/catkin_ws/src
Build space: /home/dhaval/catkin_ws/build
Devel space: /home/dhaval/catkin_ws/devel
Install space: /home/dhaval/catkin_ws/install
Creating symlink "/home/dhaval/catkin_ws/src/CMakeLists.txt" pointing to "/opt/ros/noetic/share/catkin/cmake/toplevel.cmake"
####
#### Running command: "cmake /home/dhaval/catkin_ws/src -DCATKIN_DEVEL_PREFIX=/home/dhaval/catkin_ws/devel -DCMAKE_INSTALL_PREFIX=/home/dhaval/catkin_ws/install -G Unix Makefiles" in "/home/dhaval/catkin_ws/build"
####
-- The C compiler identification is GNU 9.3.0
-- The CXX compiler identification is GNU 9.3.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Using CATKIN_DEVEL_PREFIX: /home/dhaval/catkin_ws/devel
-- Using CMAKE_PREFIX_PATH: /opt/ros/noetic
-- This workspace overlays: /opt/ros/noetic
-- Found PythonInterp: /usr/bin/python3 (found suitable version "3.8.10", minimum required is "3")
-- Using PYTHON_EXECUTABLE: /usr/bin/python3
-- Using Debian Python package layout
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Using CATKIN_DEVEL_PREFIX: /home/dhaval/catkin_ws/devel
-- Using CMAKE_PREFIX_PATH: /opt/ros/noetic
-- This workspace overlays: /opt/ros/noetic
-- Found PythonInterp: /usr/bin/python3 (found suitable version "3.8.10", minimum required is "3")
-- Using PYTHON_EXECUTABLE: /usr/bin/python3
-- Using Debian Python package layout
-- Found PY_em: /usr/lib/python3/dist-packages/em.py
-- Using empy: /usr/lib/python3/dist-packages/em.py
-- Using CATKIN_ENABLE_TESTING: ON
-- Call enable_testing()
-- Using CATKIN_TEST_RESULTS_DIR: /home/dhaval/catkin_ws/build/test_results
-- Forcing gtest/gmock from source, though one was otherwise available.
-- Found gtest sources under '/usr/src/googletest': gtests will be built
-- Found gmock sources under '/usr/src/googletest': gmock will be built
-- Found PythonInterp: /usr/bin/python3 (found version "3.8.10")
-- Found Threads: TRUE
-- Using Python nosetests: /usr/bin/nosetests3
-- catkin 0.8.10
-- BUILD_SHARED_LIBS is on
-- BUILD_SHARED_LIBS is on
-- Configuring done
-- Generating done
-- Build files have been written to: /home/dhaval/catkin_ws/build
####
#### Running command: "make -j5 -l5" in "/home/dhaval/catkin_ws/build"
####
dhaval@dhaval-VirtualBox:~/catkin_ws$
```

We need to setup new.*sh file and command is given below.

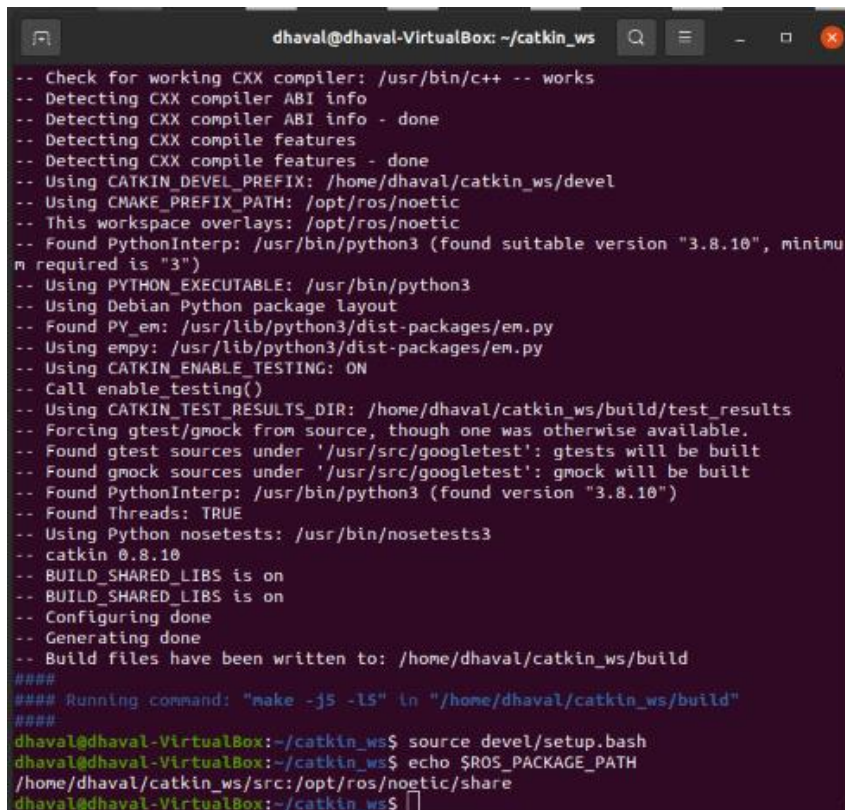
```
$ source devel/setup.bash
```

A terminal window titled 'dhaval@dhaval-VirtualBox: ~/catkin_ws' showing the output of the catkin setup process. The window has a dark background with light-colored text. The output shows various checks for compilers, Python version, and the final configuration of the catkin workspace. The process ends with the command 'source devel/setup.bash' being executed.

```
dhaval@dhaval-VirtualBox: ~/catkin_ws
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Using CATKIN_DEVEL_PREFIX: /home/dhaval/catkin_ws/devel
-- Using CMAKE_PREFIX_PATH: /opt/ros/noetic
-- This workspace overlays: /opt/ros/noetic
-- Found PythonInterp: /usr/bin/python3 (found suitable version "3.8.10", minimum required is "3")
-- Using PYTHON_EXECUTABLE: /usr/bin/python3
-- Using Debian Python package layout
-- Found PY_em: /usr/lib/python3/dist-packages/em.py
-- Using empy: /usr/lib/python3/dist-packages/em.py
-- Using CATKIN_ENABLE_TESTING: ON
-- Call enable_testing()
-- Using CATKIN_TEST_RESULTS_DIR: /home/dhaval/catkin_ws/build/test_results
-- Forcing gtest/gmock from source, though one was otherwise available.
-- Found gtest sources under '/usr/src/gtest': gtests will be built
-- Found gmock sources under '/usr/src/gtest': gmock will be built
-- Found PythonInterp: /usr/bin/python3 (found version "3.8.10")
-- Found Threads: TRUE
-- Using Python nosetests: /usr/bin/nosetests3
-- catkin 0.8.10
-- BUILD_SHARED_LIBS is on
-- BUILD_SHARED_LIBS is on
-- Configuring done
-- Generating done
-- Build files have been written to: /home/dhaval/catkin_ws/build
####
#### Running command: "make -j5 -l5" in "/home/dhaval/catkin_ws/build"
####
dhaval@dhaval-VirtualBox:~/catkin_ws$ source devel/setup.bash
dhaval@dhaval-VirtualBox:~/catkin_ws$
```

To make sure your workspace is properly overlayed by the setup script, make sure `ROS_PACKAGE_PATH` environment variable includes the directory you are in.

```
$ echo $ROS_PACKAGE_PATH  
/home/youruser/catkin_ws/src:/opt/ros/kinetic/share
```

A terminal window titled 'dhaval@dhaval-VirtualBox: ~/catkin_ws' showing the output of a catkin build. The output includes various checks for compilers, Python, and test frameworks, followed by the successful completion of the build process. The user then sources the setup script and checks the ROS_PACKAGE_PATH environment variable.

```
dhaval@dhaval-VirtualBox: ~/catkin_ws
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Using CATKIN_DEVEL_PREFIX: /home/dhaval/catkin_ws/devel
-- Using CMAKE_PREFIX_PATH: /opt/ros/noetic
-- This workspace overlays: /opt/ros/noetic
-- Found PythonInterp: /usr/bin/python3 (found suitable version "3.8.10", minimum required is "3")
-- Using PYTHON_EXECUTABLE: /usr/bin/python3
-- Using Debian Python package layout
-- Found PY_em: /usr/lib/python3/dist-packages/em.py
-- Using empy: /usr/lib/python3/dist-packages/em.py
-- Using CATKIN_ENABLE_TESTING: ON
-- Call enable_testing()
-- Using CATKIN_TEST_RESULTS_DIR: /home/dhaval/catkin_ws/build/test_results
-- Forcing gtest/gmock from source, though one was otherwise available.
-- Found gtest sources under '/usr/src/gtest': gtests will be built
-- Found gmock sources under '/usr/src/gtest': gmock will be built
-- Found PythonInterp: /usr/bin/python3 (found version "3.8.10")
-- Found Threads: TRUE
-- Using Python nosetests: /usr/bin/nosetests3
-- catkin 0.8.10
-- BUILD_SHARED_LIBS is on
-- BUILD_SHARED_LIBS is on
-- Configuring done
-- Generating done
-- Build files have been written to: /home/dhaval/catkin_ws/build
####
#### Running command: "make -j5 -l5" in "/home/dhaval/catkin_ws/build"
####
dhaval@dhaval-VirtualBox:~/catkin_ws$ source devel/setup.bash
dhaval@dhaval-VirtualBox:~/catkin_ws$ echo $ROS_PACKAGE_PATH
/home/dhaval/catkin_ws/src:/opt/ros/noetic/share
dhaval@dhaval-VirtualBox:~/catkin_ws$
```

- roscore is the first thing you should run when using ROS. To run this type run `$ roscore` command.

```
roscore http://dhaval-VirtualBox:11311/
ros-noetic-ros-tutorials is already the newest version (0.10.2-1focal.20220107.001106).
ros-noetic-ros-tutorials set to manually installed.
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi
  libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 101 not upgraded.
dhaval@dhaval-VirtualBox:~/catkin_ws$ roscore
... logging to /home/dhaval/.ros/log/6f9040b2-8b57-11ec-a4e1-23311c55160d/roslaunch-
dhaval-VirtualBox-3480.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://dhaval-VirtualBox:44269/
ros_comm version 1.15.14

SUMMARY
=====

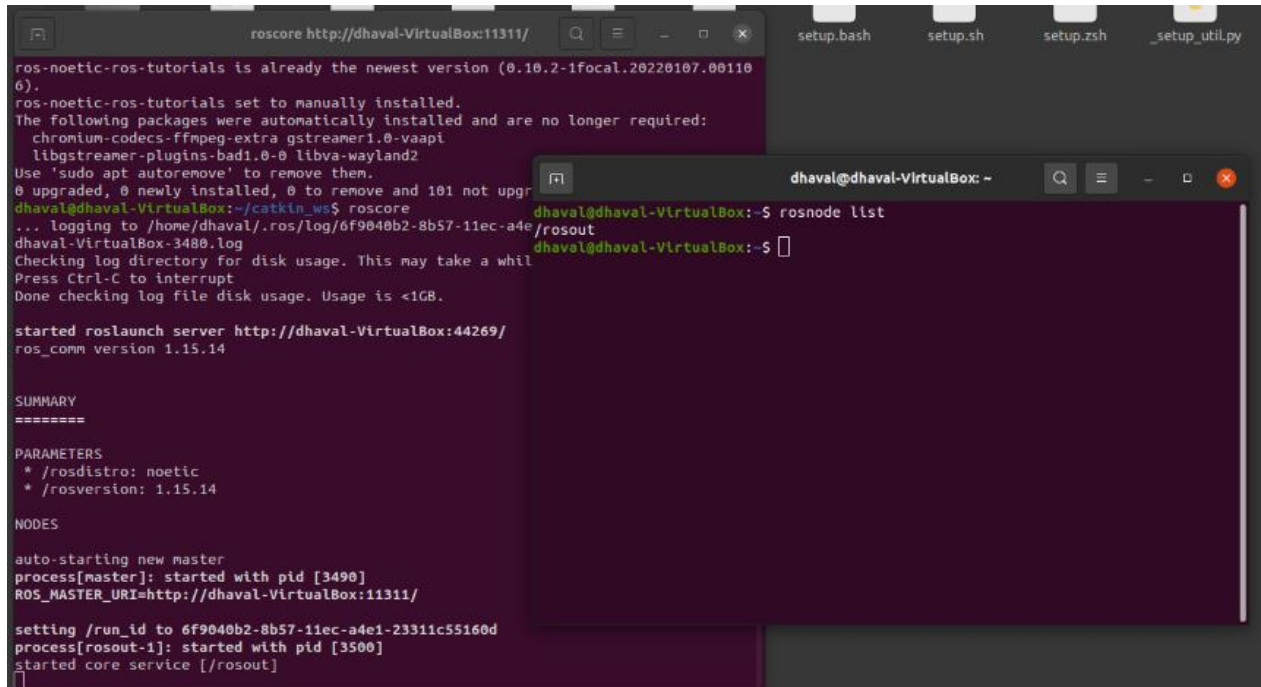
PARAMETERS
* /rostdistro: noetic
* /rosversion: 1.15.14

NODES

auto-starting new master
process[master]: started with pid [3490]
ROS_MASTER_URI=http://dhaval-VirtualBox:11311/

setting /run_id to 6f9040b2-8b57-11ec-a4e1-23311c55160d
process[rosout-1]: started with pid [3500]
started core service [/rosout]
```


- rosnod displays information about the ROS nodes that are currently running. The `$ rosnod list` command lists these active nodes:



The screenshot shows a terminal window with a dark background. The top of the window has a title bar with the text "roscore http://dhaval-VirtualBox:11311/" and several window control icons. Below the title bar, there are tabs for "setup.bash", "setup.sh", "setup.zsh", and "_setup_util.py". The main terminal area displays the output of a series of commands. It starts with a message about ROS-NOETIC-ROS-TUTORIALS being the newest version. Then, it shows the output of "roscore", which includes a list of packages that were automatically installed and are no longer required. This is followed by a message about logging to a directory and checking log file disk usage. The terminal then shows the output of "roslaunch server http://dhaval-VirtualBox:44269/". Below this, there is a "SUMMARY" section, followed by "PARAMETERS" and "NODES" sections. The "NODES" section shows the auto-starting of a new master process. Finally, the terminal shows the output of "roslaunch", which includes setting a run_id and starting a core service. A second terminal window is overlaid on the first, showing the command "roslaunch" and its output, which includes the command "roslaunch" and its output, which includes the command "roslaunch" and its output.

```
ros-noetic-ros-tutorials is already the newest version (0.10.2-1focal.20220107.001106).
ros-noetic-ros-tutorials set to manually installed.
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi
  libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 101 not upgr
dhaval@dhaval-VirtualBox:~/catkin_ws$ roscore
... logging to /home/dhaval/.ros/log/6f9040b2-8b57-11ec-a4e1-rosout
dhaval-VirtualBox-3480.log
Checking log directory for disk usage. This may take a while
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://dhaval-VirtualBox:44269/
ros_comm version 1.15.14

SUMMARY
=====

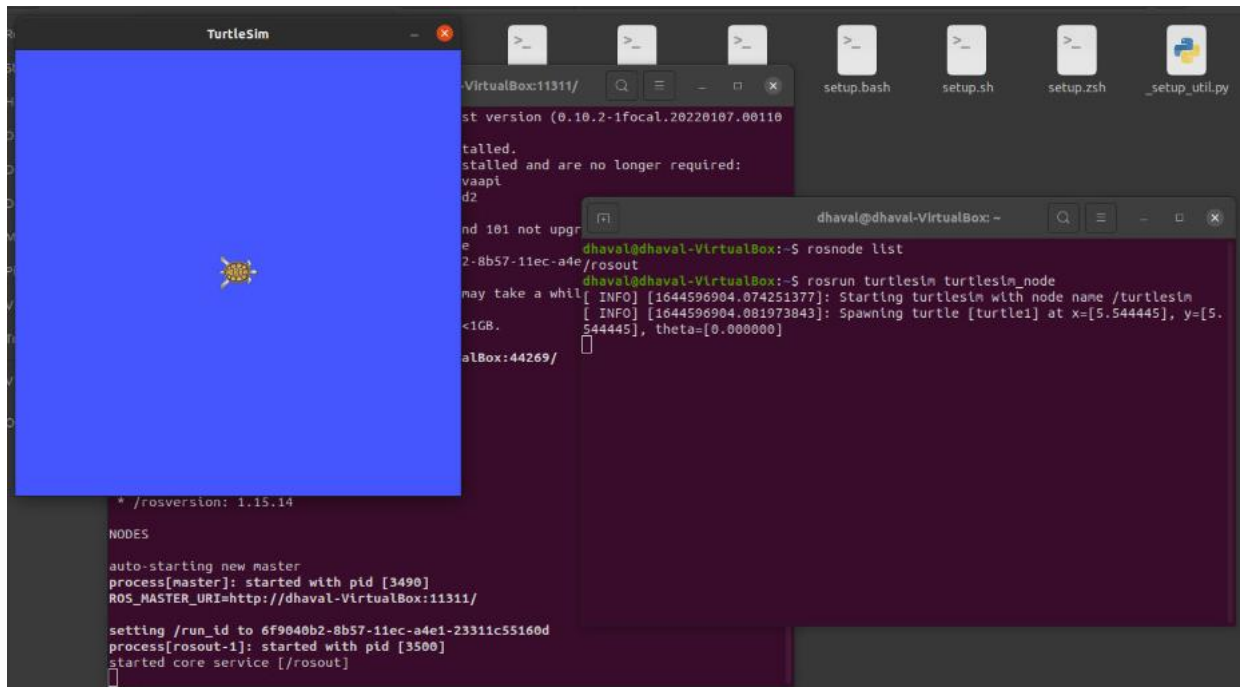
PARAMETERS
 * /rostdistro: noetic
 * /rosversion: 1.15.14

NODES
auto-starting new master
process[master]: started with pid [3490]
ROS_MASTER_URI=http://dhaval-VirtualBox:11311/

setting /run_id to 6f9040b2-8b57-11ec-a4e1-23311c55160d
process[rosout-1]: started with pid [3500]
started core service [/rosout]

dhaval@dhaval-VirtualBox:~$ roslaunch
dhaval@dhaval-VirtualBox:~$
```

- To run the `turtlesim_node` in the `turtlesim` package by using `roslaunch turtlesim turtlesim.launch` command.



- In new terminal, type command `$ rosnode list`.

